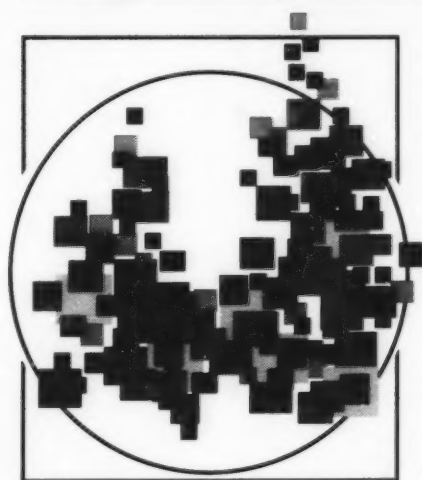
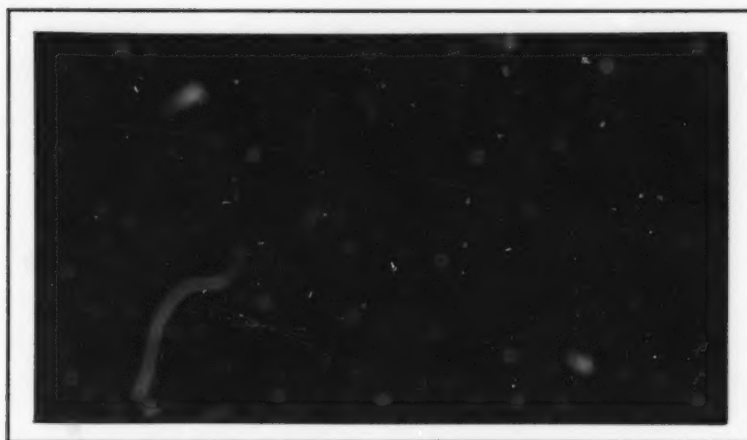




CENTRE FOR
SYSTEMS
SCIENCE

LCCR

**A New Coding Technique for
Spectral Shaping of Data**

James K. Cavers and Robert F. Marchetto

CSS/LCCR TR90-08

March 1990

A NEW CODING TECHNIQUE FOR SPECTRAL SHAPING OF DATA

James K. Cavers

Communications Science Laboratory

School of Engineering Science

Simon Fraser University

Burnaby B.C. V5A 1S6

CANADA

Tel (604) 291-3281

Robert F. Marchetto

Glenayre Electronics Ltd

1570 Kootenay St

Vancouver B.C. V5K 5B8

CANADA

Tel (604) 293-1611

ABSTRACT

Spectrum shaping codes introduce redundancy and correlation into a transmitted data stream in order to alter the transmitted spectrum without affecting the pulse shape or peak-to-average power ratio. Most are based on algebraic properties, or on a "running digital sum". This paper introduces a new and very general approach to spectral shaping, in which we employ the Viterbi Algorithm at the transmitter to minimize the transmitted power in specified bands. The principal advantage is greatly increased flexibility of width, depth and placement of spectral notches. The computational load at the transmitter is greater than that of earlier algorithms, although the receiver is unaffected.

March, 1990

I. INTRODUCTION

Spectrum shaping codes [1,2,3] add redundant bits or levels in order to alter the transmitted spectrum without changing the pulse shape. They are most commonly used to suppress signal power near DC to minimize the effect of AC coupling, and to provide sufficient data transitions for reliable clock recovery. More recently, there have been proposals [4,5,6] to use the spectral null to accommodate a pilot tone. Such codes can also suppress power at frequencies near half the symbol rate as protection against band-edge filter distortion. Examples include the well-known bipolar and duobinary codes [2,7], B6ZS [2], and binary block codes (BBCs) [8,9].

However, codes reported to date have had limitations in their spectrum shaping ability: those able to produce wide notches have required nonbinary alphabets [2,7], while binary codes result in very narrow notches [3,9]. Moreover, BBCs have been limited to creating nulls at DC and/or half the symbol rate.

The present paper introduces a binary coding technique with a number of new and desirable properties. By generalizing binary block codes, LABI (look ahead block inversion) can create spectral nulls of arbitrary shape and location, as well as shape the spectrum in other ways. It can trade depth against width and, for a given specification, minimize the fraction of redundant symbols. With complex data values it can generate asymmetric nulls, on only one side of channel center. LABI is based on an alternative formulation of the shaping problem and an unconventional use of the Viterbi Algorithm at the transmitter. Its uses include generation of a notch for a doppler spread pilot [6], improved protection against AC coupling, and protection for or against signaling tones anywhere in the band.

In exchange for its benefits, LABI demands a significantly higher level of computation at the transmitter than does a conventional BBC. However, the operation and complexity of the receiver is completely unaffected.

Section 2 of the paper reviews BBCs. Section 3 introduces LABI, and demonstrates its shaping ability. Section 4 presents some

generalizations, and the results are summarized in Section 5.

II. BINARY BLOCK CODES

A. General Structure

The transmitter prepares a BBC by partitioning the input data bits $x(n)$ into blocks of length N'_b and appending a flag bit of value +1 to each one, making blocks of length $N_b = N'_b + 1$. The blocks, with the flag bit, are then inverted in polarity or left intact, as appropriate, to achieve the desired spectral shaping:

$$z(iN_b + j) = F(i) y(iN_b + j) \quad (1)$$

where $y(n)$ is the data sequence after padding with flag bits, $F(i)$ is the polarity of the i^{th} block, and i and j are the block index and the bit index within a block, respectively. The receiver performs the operations in reverse, inverting any block received with its flag bit equal to -1. Figure 1 shows the format of the transmitted data stream. Extension to QPSK and multilevel codes is straightforward, and is described in Section IV.

The transmission efficiency of a BBC is given by:

$$\eta = (N_b - 1)/N_b \quad (2)$$

Clearly, it results in an increase in channel transmission rate and bandwidth, or a reduction in effective data rate. It also increases bit error rate by the same factor as does differential encoding (i.e. an approximate factor of 2): if the channel bit error probability is p , then the error probability of the data bits is p if the flag bit is received correctly, and $1-p$ if the flag bit is received in error. The average BER for released bits is therefore:

$$\text{BER} = 2p(1-p) \approx 2p \quad (3)$$

Note that the transmitted power spectrum is a product of spectra due to the discrete-time coded bit sequence and the pulse shape. Since they can be treated separately, we consider only the spectrum of the bit sequence.

B. RDS Binary Block Code

The essence of any BBC is the method of determining the sequence of

block polarities. Since $F(i)$ is also the polarity of the flag bits, we will term it the flag sequence. The original BBC [8,9], which we call RDS BBC (for running digital sum) operates as follows. It calculates a running interference sum for each block following the appending of the flag bit:

$$RIS(i) = \sum_{j=0}^{N_b-1} y(iN_b+j) \quad (4)$$

and a running digital sum up to block r :

$$RDS(r) = \sum_{i=-\infty}^r F(i)RIS(i) \quad (5)$$

The block polarity $F(i)$ is selected as:

$$F(i) = \begin{cases} 1, & \text{if } RIS(i)RDS(i-1) \leq 0 \\ -1, & \text{otherwise} \end{cases} \quad (6)$$

In other words, a block is inverted if its RIS has the same sign as the accumulated RDS up to the previous block; otherwise, it is left unchanged. This algorithm limits the magnitude of the RDS to N_b , and therefore produces a spectral null at DC. A variation [7] produces a notch at the Nyquist frequency $1/2T$, where T is the symbol duration.

The attractions of the algorithm are its simplicity and the infinite attenuation in the center of its notch. Figure 2 shows typical spectra. However, the notch is very narrow, and the only way to widen it is by decreasing the block length, with a corresponding sacrifice in efficiency. This is unsatisfactory for many applications. One example is AC coupling in which the highpass cutoff frequency is a significant fraction of the symbol rate. Another [4,6] is creation of a notch for a doppler-spread pilot tone, in which the doppler frequency is a significant fraction of the symbol rate.

III. LOOK AHEAD BLOCK INVERSION

Look ahead block inversion (LABI) is a BBC. However, the flag bits are determined with a different and more general algorithm. We start with a desired shape $S(f)$ of the data power spectrum. Figure 3 illustrates the simple case of a wide notch centered at DC. We also define the corresponding filter:

$$H(f) = \sqrt{S(f)} \quad (7)$$

If the contiguous data blocks, including the flag bits, are passed through the filter $H(f)$, then the output power is a measure of the degree to which the data spectrum resembles $S(f)$. The flag bits are chosen so that the power in this output is maximized, and the spectral density is therefore made to approximate $S(f)$.

In an alternative (though equivalent) view, we can define a complementary filter $H_c(f)$:

$$H_c(f) = \sqrt{S_m - S(f)} \quad (8)$$

where S_m is the maximum value attained by $S(f)$. When the contiguous data block are passed through $H_c(f)$, its output represents an unwanted residual signal. As an example, in AC coupling the complementary filter is a lowpass, and its output is the linear distortion, or intersymbol interference (ISI), suffered by the data signal. Choosing the flag bits to minimize the power at the output of $H_c(f)$ minimizes the mean square ISI.

This is quite a general formulation of the spectrum shaping problem. The filters can specify arbitrary spectra, typically (though not necessarily) notches of various widths and locations, and the filters are not required to be causal. It includes the original RDS BBC as a special case, in which the unit pulse response $h_c(k)$ is the unit step.

B. The Minimization Problem

After appropriate windowing, the complementary filter has a finite impulse response $h_c(n)$ which spans $N_f = 2L+1$ blocks of N_b coefficients per block, for a total of $N_c = N_f N_b$ coefficients. For simplicity, the

treatment below assumes that it is centered about $n=0$. If the filter is causal, then the coefficients are zero for $n < 0$.

For a given flag bit sequence, the coder calculates the complementary filter output by the convolution:

$$e(iN_b + j) = \sum_{k=-L}^{L-1} \sum_{m=0}^{N_b-1} F(i-k) y((i-k)N_b + m) h_c(kN_b + j - m) \quad (9)$$

where i and k index the blocks and j and m index within a block. Our RIS is defined as the total squared output over one block length:

$$RIS(i) = \sum_{j=0}^{N_b-1} e^2(iN_b + j) \quad (10)$$

The quantity to be minimized with respect to the flag sequence $F(i)$ up to block r is:

$$J(r) = \sum_{i=-\infty}^r RIS(i) \quad (11)$$

Since each flag bit affects the RIS of both past and future blocks, determining the sequence which minimizes $J(\infty)$ appears to involve an exhaustive search. Fortunately, however, this is precisely the problem solved by the well-known Viterbi Algorithm [10], though our use of it is unconventional.

C. Viterbi Algorithm Formulation

Since the filter $h_c(n)$ spans only $N_f = 2L+1$ blocks, its output in the center of a block (see (9)) is determined by only N_f flag bits. At other bit positions within a block, its output is determined in principle by N_f+1 flag bits. However, for simplicity, we window the input signal to N_f contiguous blocks, so that $RIS(i)$ is determined by only N_f flag bits. We define the state vector:

$$s(i) = \langle F(i-L), \dots, F(i), \dots, F(i+L-1) \rangle \quad (12)$$

and note that it can take on only a finite number of values:

$$s_n, n = 0, \dots, 2^{N_f-1} \quad (13)$$

Given the flag-padded data sequence $y((i-k)N_b+m)$, $k=-L, \dots, L-1$, the RIS at block i is determined only by the state transition $\langle s(i), s(i+1) \rangle$. We change notation slightly and denote it as $\text{RIS}(i, s(i), s(i+1))$. A trellis can be defined in the usual way, with RIS serving as the branch metric. The problem then becomes finding the allowable state sequence (path through the trellis) which minimizes:

$$J(\infty) = \sum_{i=-\infty}^{\infty} \text{RIS}(i, s(i), s(i+1)) \quad (14)$$

Thus the Viterbi Algorithm applies directly. Figure 4 shows the organization of the resulting LABI encoder.

One peculiarity is that, because of the quadratic metric, the entire sequence of flag bits can be inverted in sign without changing J . Thus there are two minimizing paths through the trellis. This condition can be eliminated, and the number of states reduced by a factor of 2, if the flag bits are differentially encoded and the states defined in terms of differentially encoded values.

D. LABI Spectra

It is clear that an analytical determination of spectrum is unlikely for a scheme of this complexity. In fact, even for the simpler RDS BBC, only very limited results have been obtained [9], and they serve primarily to reduce the length of simulation runs. We have had to resort to simulation exclusively to obtain power spectra.

Figure 5 shows a LABI spectrum obtained with $S(f)$ specifying a notch 27 dB deep from 0 to 2% of the symbol rate, and a transition band from 2% to 4%. The complementary filter $h_c(n)$ has 55 coefficients. As N_b drops from 11 to 5 bits, the resulting spectrum approaches $S(f)$, though the efficiency drops from 0.91 to 0.8. Comparison with Figure 2 shows that RDS BBC produces a much narrower, though deeper, notch. Figure 6 demonstrates that decreased width requirements allow deeper notches in LABI, as well. Figure 7 illustrates LABI's ability to generate notches at any point in the spectrum, in this case centered at 24% of the symbol rate.

Despite the lack of analytical results, rough guidelines are possible regarding parameter choices and the degree to which the actual

spectrum approximates $S(f)$. The quality of the approximation increases as the ratio of correlation time of $h_c(n)$ to the block length N_b increases, since the algorithm, in effect, makes a staircase approximation to $h_c(n)$. If the objective is a notch of a given shape, then the correlation time varies inversely with notch width, because of the usual inverse time-frequency scaling in Fourier analysis. For a given quality of spectrum, therefore, both N_b and N_c vary inversely with notch width, and their ratio N_f is relatively insensitive to width.

The actual notch depth achieved by LABI is determined only partly by the desired depth specified in the stopband attenuation of $H_c(f)$. As the desired attenuation increases, so does the length N_c of $h_c(n)$, and N_f with it. However, increasing the length quickly runs into diminishing returns, as the actual notch depth becomes limited by the ratio of correlation time of $h_c(n)$ to N_b , as discussed above. On the other hand, if N_c and N_b are fixed, then the notch attenuation can be traded against notch width, through variation in the their ratio N_f .

D. Computational Requirements

The computational effort in LABI is dominated by the branch metric calculations. If each RIS value is calculated by explicit convolution, (9) and (10) show that the effort (the number of multiply-accumulate operations) is $N_c N_b$. The load per block is therefore given approximately by the load per RIS calculation times the number of state transitions, and the computational load per data bit is:

$$C = \frac{1}{2} N_c \frac{N_b}{N_b - 1} 2^{N_f} \quad (\text{explicit convolution}) \quad (15)$$

if the flag bits are differentially encoded. The effect of more stringent spectral shape requirements, as noted above, is to increase N_c and N_f , and decrease N_b . The result is a higher computational load and a lower efficiency. One mitigating factor is that the exponent N_f is relatively insensitive to notch width, as noted earlier, though it does increase with increasing depth requirements.

Calculation of the branch metrics $\text{RIS}(i, s(i), s(i+1))$ is simplified in the binary case by table lookup. From (10), we have to calculate the complementary filter output $e(in_b + j)$ at each position j of block i .

From (9), this value can be obtained as the sum of contributions from N_f adjacent blocks, each of N_b bits. The N_b-1 independent bits in each block can be used as an index into a table of contributions. Since there are N_b such positions j , and N_f tables for each position, the required table space is:

$$N_t = N_b N_f 2^{N_b-1} \quad (16)$$

The load in calculating a single RIS branch metric therefore consists of determining the binary values $F(i-k) y((i-k)N_b+m)$ in each of the N_f relevant blocks (i.e. invert block contents if $F(i-k)=-1$), looking up N_f tables and summing the results, squaring the sum, and repeating for the other N_b-1 positions in block i . The load per block therefore equals $N_f N_b$, if table lookup and addition is given the same weight as a multiply-accumulate operation. The RIS must be computed for each allowable transition, so that the total load per data bit, similar to that in (15), is:

$$C = \frac{1}{2} N_f \frac{N_b}{N_b-1} 2^{N_f} \quad (\text{table lookup}) \quad (17)$$

This is a significant reduction.

As an example, the consider the spectra in Figure 5. They were produced with $N_c = 55$. For $N_b = 11$, we have $N_f = 5$. There are then 8 states, and a computational load per data bit of 968 operations (explicit convolution) or 88 operations (table lookup). For $N_b = 5$, there are 512 states, and a load of 70,400 operations (explicit convolution) or 14,080 operations (table lookup). This is, of course, much greater than in the RDS BBC, and is the price paid for better spectral shaping.

IV. GENERALIZATIONS OF LABI

A. Multilevel Modulation

To this point, we have considered only binary modulation. However, LABI also extends to multilevel modulation. Assume that the symbols have L amplitude levels:

$$a_k = k - \frac{L-1}{2}, \quad k = 0, \dots, L-1. \quad (18)$$

There are at least two straightforward generalizations to multiple amplitudes. The first follows directly from the basic BBC, as proposed in [4]: simply append a known symbol to each block, and invert blocks as required, just as in the binary case. The number of states and the computational load for explicit convolution are unchanged from the binary case with the same block size. However, it dramatically increases the memory space required for table lookup computation.

A second generalization is to add a constant modulo L to each symbol in a block. That is, if a symbol before coding has the value:

$$y(iN_b + j) = a_{I(i,j)} \quad (19)$$

then coding converts the index $I(i,j)$ to a new value:

$$I'(i,j) = (I(i,j) + F(i)) \bmod L \quad (20)$$

where $F(i)$, the additive constant for block i , is an integer in $0 \dots L-1$. This method gives more flexibility than simple block inversion, though it reduces to block inversion in the binary case. The price is significantly increased computation, since the number of state transitions is now $\exp_L(N_f-1)$ instead of $\exp_2(N_f-1)$.

B. Complex Symbols

If the signal is to be modulated on a carrier, then the data symbol values are, in general, complex. LABI extends trivially to QAM such as QPSK and 16QAM: the real and imaginary components are processed separately and independently, as described above for real signals.

In the case of PSK with more than 4 phases, the output $e(n)$ of the complementary filter $h_c(n)$ is complex. The metric (10) is slightly modified:

$$RIS(i) = \sum_{j=0}^{N_b-1} |e(iN_b+j)|^2 \quad (21)$$

and the LABI alteration to the symbols of a block consists of rotation by a multiple of $2\pi/L$, where L is the number of phases. If L is even, and the rotations are restricted to 0 and π , this method reduces to the original LABI block inversion scheme.

Once in the complex domain, another interesting generalization is possible. For a mere doubling of computational effort, the coefficients of filter $h_c(n)$ can be complex, which allows $H_c(f)$ and the desired spectral shape $S(f)$ to be asymmetric with respect to channel center. Thus we can use LABI to create a notch on only one side of the channel, or to perform other asymmetric shaping.

C. Alternative Metrics

To this point, we have considered only the spectra produced by LABI with a quadratic metric. If it is used to control intersymbol interference, though, LABI may fare better with a different metric. Consider AC coupling, in which the output $b(n)$ of an appropriately chosen $h_c(n)$ represents the ISI. In a Gaussian noise environment, the error rate depends exponentially on the square of the distance between the decision boundary and the ISI-shifted signal, and so is dominated by the worst ISI values. A more appropriate metric for reduction of error rate may be:

$$RIS(i) = \sum_{j=0}^{N_b-1} |e(iN_b+j)|^p \quad (22)$$

where p is greater than 2, or its equivalent for large p :

$$RIS(i) = \max_j \sum_{j=0}^{N_b-1} |e(iN_b+j)| \quad (23)$$

This is an interesting issue for further investigation.

D. FSK

Although LABI is easiest to apply to the linear modulations considered to this point, it can also be adapted to some nonlinear

modulations, such as FSK. The simplest case is MSK (minimum shift keying), that is, FSK with modulation index $1/2$. MSK is well known [11] to be equivalent to offset QPSK with half-sine pulse shaping. Thus LABI can be applied independently to the real and imaginary components of the data in order to produce MSK with additional spectral notches.

It is tempting to extend this approach to include other continuous phase modulation (CPM) signals such as GMSK and TFM. Many of them have an approximate representation as offset QAM, in which the spectrum could be manipulated with a BBC, as was done in [5]. However, these approximate representations do not have a constant envelope, and are consequently of lesser interest for transmitter applications than the original CPM.

It is also possible, of course, to create a notch in binary data by applying LABI before the CPM step. Although this does not create a corresponding null in the transmitted signal, it still circumvents AC coupling problems when the signal is detected differentially or by discriminator.

V. CONCLUSIONS

Effective spectral shaping can be achieved by manipulating the polarity of transmitted data blocks, with no change to pulse shape or peak to average power ratio. The art lies in the strategy by which block polarities are chosen. Our new algorithm, LABI (look ahead block inversion), allows more flexible spectral distribution of power than an earlier algorithm, which we call RDS BBC.

LABI is based on use of the Viterbi Algorithm at the transmitter to minimize the weighted power in different spectral regions. This formulation contains the seeds of many generalizations, such as multilevel modulation, asymmetric shaping of bandpass signals, minimax rather than mean square minimizations, and MSK.

The computational load of LABI depends on notch specifications, but is significantly higher than that of its predecessor RDS BBC. However, this additional complexity is required only at the transmitter, which in many modem implementations does not carry a heavy computational load. The receiver, which normally carries the heavier load, is unchanged from RDS BBC.

REFERENCES

- [1] H. Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data", IEEE Trans Commun Tech, vol COM-19, no 6, pp 1067-1100, December 1971.
- [2] A. Croiser, "Introduction to Pseudoternary Transmission Codes", IBM J Res Develop, vol 14, pp 354-367, July 1970.
- [3] E. Gorog, "Redundant Alphabets with Desirable Frequency Spectrum Properties", IBM J Res Develop, vol 12, pp 234-241, May 1968.
- [4] D.Y. Kim and K. Feher, "New Carrier and Symbol Synchronization Technique for Digital Mobile Systems", Proc IEEE Vehic Tech Conf, pp 371-376, Philadelphia, May 1988.
- [5] P.S.K. Leung and K Feher, "Transparent Tone in Band (TTIB) Aided GMSK/MSK Modem Systems", Proc IEEE Vehic Tech Conf, pp 249-255, San Francisco, May 1989.
- [6] J.K. Cavers, R.F. Marchetto and S.D. Carlson, "A New Spectral Notch Generator for Pilot Tone Systems", Proc IEEE Vehic Techn Conf, Orlando, May 1990.
- [7] D.Y. Kim and J-K Kim, "A Condition for Stable Minimum-Bandwidth Line Codes", IEEE Trans Commun, vol COM-33, no 2, pp 152-157, February 1985.
- [8] F.K. Bowers, "Pulse Code Transmission System", US Patent No. 2,957,947, Oct 25, 1960.
- [9] L.J. Greenstein, "Spectrum of a Binary Signal Block Coded for DC Suppression", Bell Syst Tech J, vol 53, no 6, pp 1103-1126, July-August, 1974.
- [10] J.G. Proakis, Digital Communications, McGraw-Hill, 1983.
- [11] S. Pasupathy, "Minimum Shift Keying: a Spectrally Efficient Modulation", IEEE Comms Mag, pp 14-22, July, 1979.

ILLUSTRATIONS

- Figure 1: Format of Transmitted BBC Data
- Figure 2: Power Spectral Density of RDS BBC
- Figure 3: Typical LABI Specification Filters
- Figure 4: Organization of the LABI Transmitter
- Figure 5: Power Spectral Density of LABI Data - Wide Notch
- Figure 6: Power Spectral Density of LABI Data - Narrow Notch
- Figure 7: Power Spectral Density of LABI Data - Nonzero Centered Notch

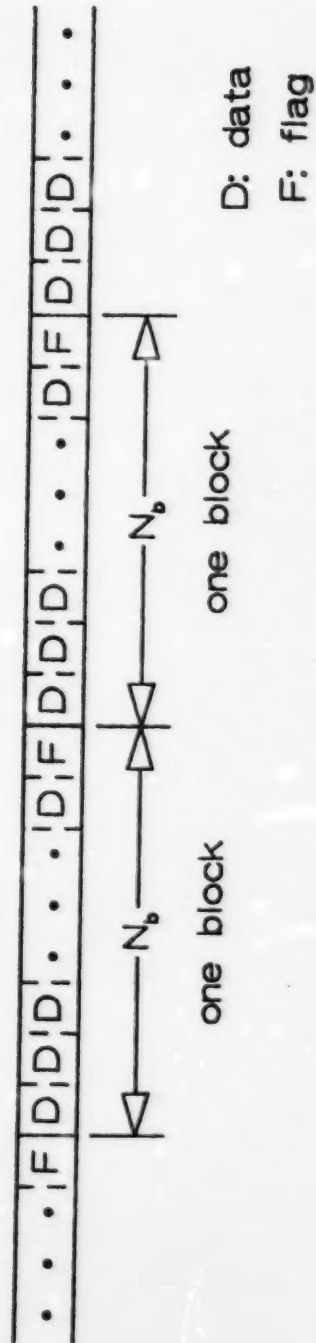


Figure 1: Format of Transmitted BBC Data

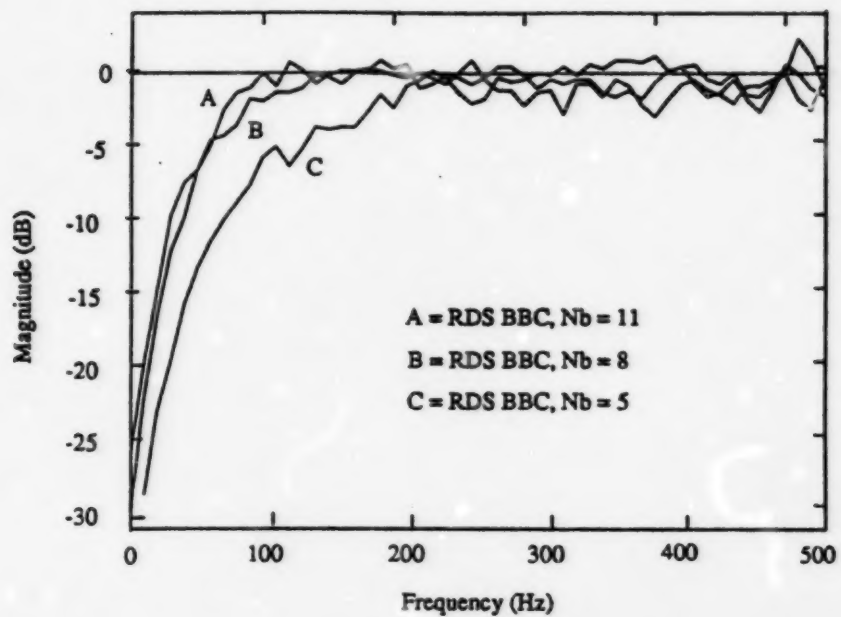


Figure 2: Power Spectral Density of RDS BBC

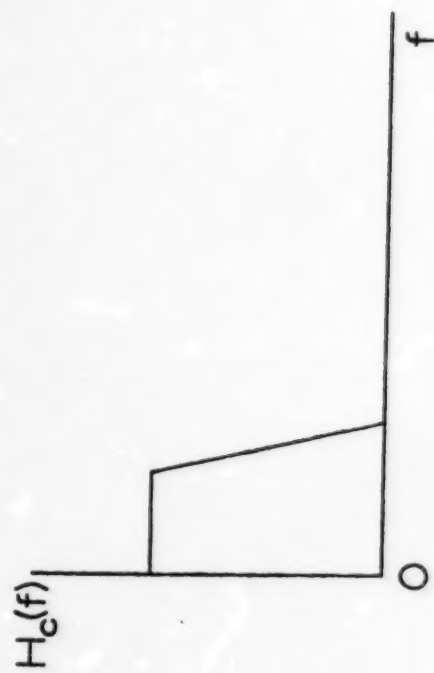


Figure 3: Typical LABI Specification Filters

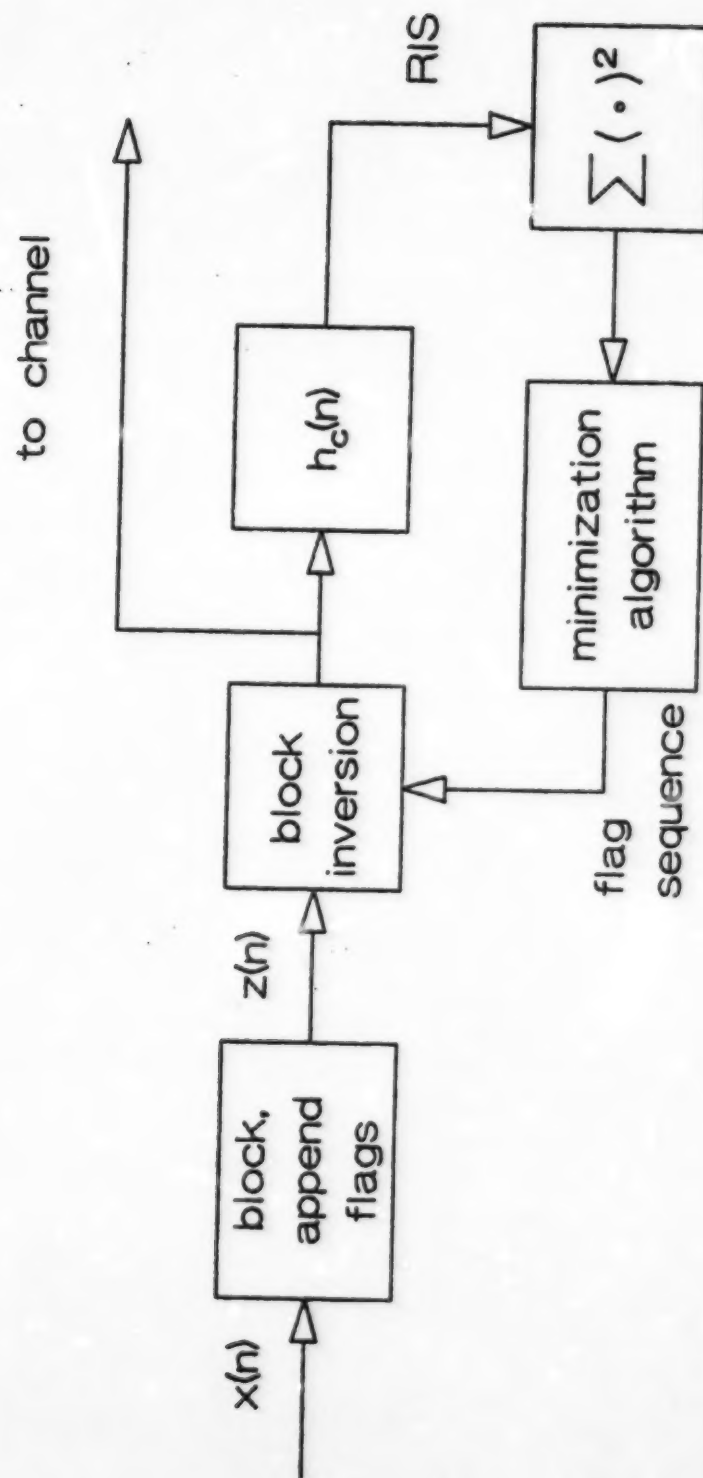


Figure 4: Organization of the LABI Transmitter

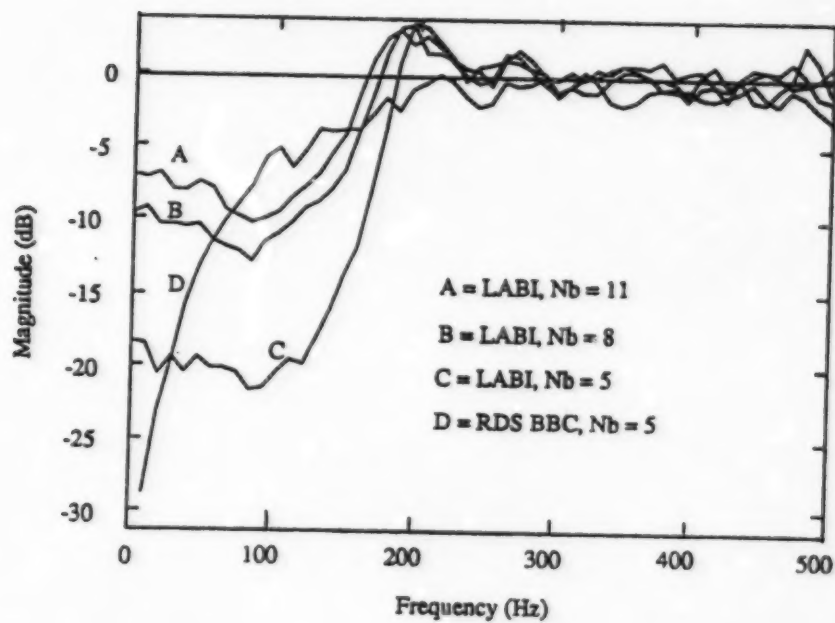


Figure 5: Power Spectral Density of LABI Data - Wide Notch

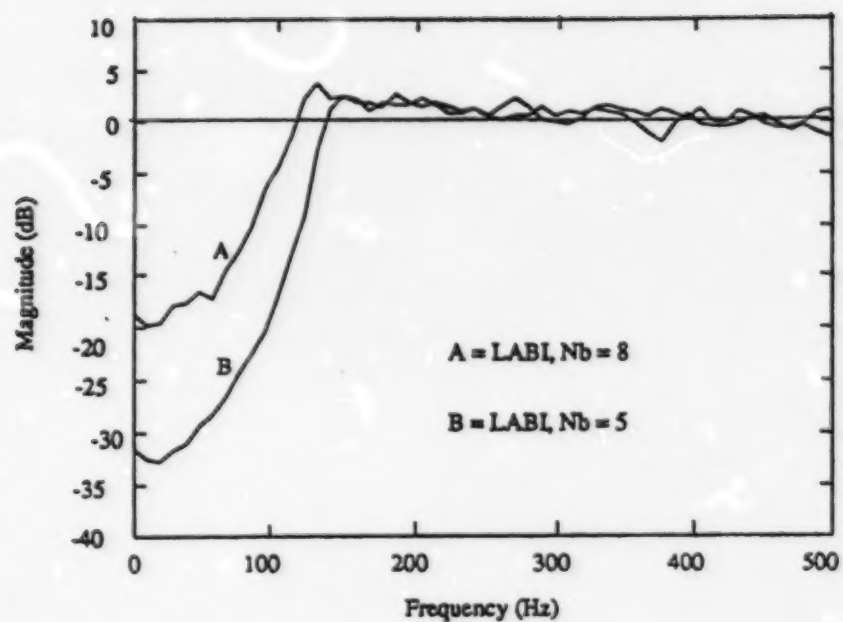


Figure 6: Power Spectral Density of LABI Data - Narrow Notch

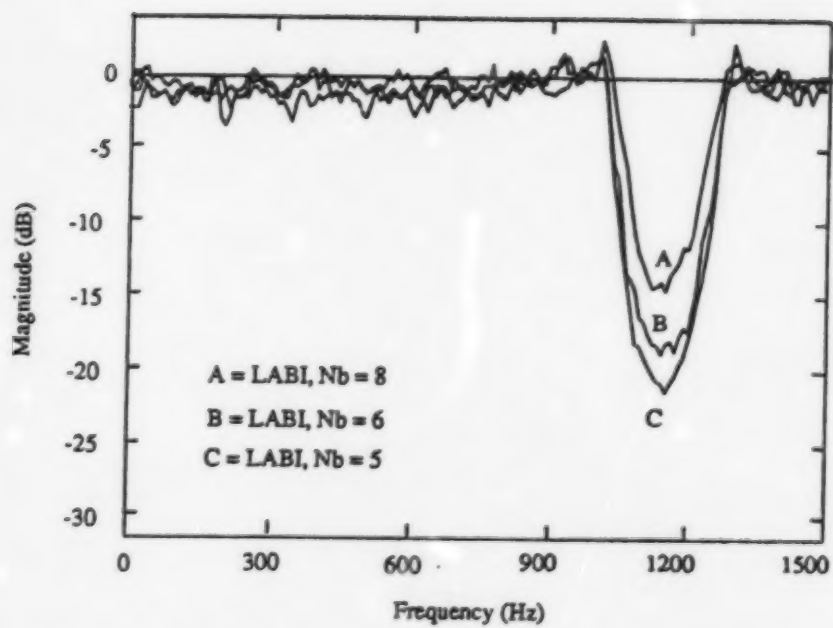


Figure 7: Power Spectral Density of LABI Data - Nonzero Centered Notch

Centre for Systems Science

- Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.
- CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.
- CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.